III. REMARKS

A. References

6. The office action relied on the following references:

- U.S. Patent Application Publication 2002/0092004, John Michael Lee, et al., entitled "Methods and systems for automatically generating software applications," filed July 26, 2001, ("Lee")
- U.S. Patent Application Publication 20020129096, Peter M. Mansour, et al., entitled "Platform-independent distributed user interface client architecture," filed February 14, 2001, ("Mansour")
- U.S. Patent Application Publication 2004/0015476, Graham Kennedy Twaddle, entitled Method and system for dynamic web-page generation, and computer-readable storage," filed under PCT August 31, 2001, ("Twaddle")

B. Overview of Office Action

7. The office action:

- Objected to the specification for failing to provide proper antecedent basis for "internet service provider" in claim 19.
- Objected to claims 1 and 5 for informalities.
- Rejected claims 1-4, 6-7, 9-14, 17-18, and 20 as being anticipated by Lee.
- Rejected claim 5 as being obvious in view of Lee in further view of Twaddle.
- Rejected claims 8 and 19 as being obvious in view of Lee in further view of Mansour.
- Rejected claims 15 and 16 as being obvious in view of Lee in further view of Mansour and Twaddle.

8. Reconsideration of this application, in view of the following remarks, is respectfully requested.

IV. SPECIFICATION AMENDMENTS

9. The specification has been amended as required to provide explicit antecedent for Internet service providers. This amended was required by the office action and adds no new matter as the subject matter was included in the original claims.

V. CLAIM AMENDMENTS

10. The claims were amended as required to correct informalities. This amended was required by the office action and adds no new matter. These claim amendments were not made to overcome prior art.

VI. OVERVIEW OF THE PRESENT INVENTION

11. The present invention is directed to a system for rapidly developing database driven web sites using a database management system, in particular an Oracle database. The system is implemented as a fully functional web site with most of the features required by all web sites.

12. The system includes a number of existing object modules that perform a variety of complex functions typically found in any web application. These include:

- User registration/membership
- User roles
- Secure access to the application, login and "forgot my password" services
- Sessions without the use of cookies
- Functions and privileges

- Dynamic menus (based on user role, logged in, logged out), including bulletin and message notification
- Administrative tools to allow for web site management
- Site-wide bulletins and broadcasts
- Inter user messaging
- E-mail
- Writing files to disk
- E-commerce functions that include a store, shopping cart, invoicing, checkout, credit card payment processing and receipts

13. The present invention provides for rapid development because:

- preexisting, preprogrammed functional elements as part of working web site
- code generators for prototyping and final working code
- interface layer is separate and facilitates a centralized, single point look and feel change
- document generator will generate system documentation (entity, abbtributes, buiness rules)

14. Other features of the present invention include:

- Pages are dynamically generated and never stored. This allows for not only browser specific functionality but also user specific functionality
- The code architecture is structured in 3 layers

VII. OVERVIEW OF LEE

15. Lee describes a generic, all encompassing, general purpose database driven application builder. He has also included design methodology. He is attempted to cover many areas including:

- data modeling (defining entities, attributes, relationships) with methodology similar to that written by Richard Barker "Oracle CASE Method" 1990-1992
- automatic generation which includes
  - o code generation for windows executables (e.g. VB) and for the web (HTML, JavaScript, etc.) using theme selections, etc.
  - o database table generation
  - o directory creation, etc.
  - o generates static and database driven dynamic applications
- post generation customization through parameters

A. Distinctions over Lee

16. There are many distinctions between what is taught by Lee and the present invention. For example, in one embodiment of the present invention, the generated code is compiled and then dynamically creates HTML and WML pages. This dynamic generation makes decisions that take into account parameters such as requesting browser on a computer or a phone, PDA, etc. It can also consider data such as user, company and other preferences. All of these can have an effect on the output dynamically produced. Lee does not do this. In addition, it appears that the platform(s) chosen by Lee would not support such functionality.

VIII. REJECTIONS BASED ON ANTICIPATION BY LEE

17. As will be discussed in detail in the following sections, Lee does not anticipate nor render obvious claims 1 through 20.

A. Claim 1(a)

"A system for developing and maintaining a network based application wherein said application has a common look and feel, said system comprising:

a. a database comprising data tables and storage..."

18. Claim 1 requires one fully functional application web site with a robust database "The system of the present invention is designed for use with a robust, industrial strength database. Oracle, for example, is portable to many platforms." ([0098]:1-3). In addition "the toolkit comes complete with an initial working database model (that manages those functions common to most web sites) and the web site user interface pages including graphics, code and pre-programmed modules that can be used as is or customized. They are included in the project and are fully functional. ([0036]:13-18). See also: [0085]-[0097].

19. In contrast, Lee Fig. 1 teaches a variety of databases including "a design database 30 and generated system database 32:" ([0034]:3-4) An application meta model must be defined in the design database to facilitate application design. The application or "generated system database" is maintained separately. Both of these databases are neither pre-populated nor fully functional as an "application web site".

B. Claim 1(b)

"b. a code generator interfacing with said database"

20. Claim 1 requires a code generator which uses a code definition file (e.g. a marked up create table statements file) and generates database source code (for example

PL/SQL) for the application layer. "First the developer must create a code definition file 160. This file is used as input to the code generator 170, which, in the Oracle embodiment, produces "standardized" Oracle PL/SQL source code."([0122]:2-6)

21. In contrast, Lee teaches "a design application configured to receive a system design and create a design database file, and a generator application configured to receive the design database file and create a computer-generated software application that includes a presentation tier, a business tier and a data tier." ([0009]:4-9). While Lee teaches the capture of some data with the designer, the data captured is not entirely the same. There are also differences in Lee's implementation.

C. Claim 1(c)(i)

"c. toolkit programs, stored in said database, comprising:

i. an application layer..."

22. The present invention requires "application layer defines the operation of the application" (Claim 17). There is one set of packages for each function in the application. Not all functions map to entities. The functions are written in PL/SQL. No presentation code (look and feel) is generated as part of this generated code.

23. In contrast, Lee teaches, "In Step 245, the generator program 28 generates the presentation tier of the application. In this step, the presentation controller classes 162 are created for each entity 50 needed for the generation application including, in a preferred embodiment, security and reports. In Step 250, the generator program 28 creates the business object code 164 for each entity 50 used in the generated software application 40." ([0065]:12-19) With presentation controller classes are generated for each entity, the presentation in whole or in part is included per entity and not implemented separately.

D. Claim 1(c)(ii)

"ii. an interface layer, and ..."

24. The present invention requires an "interface layer defines a plurality of looks and feels whereby one look and feel can be changed though out the entire application without changing the remaining looks and feels and without changing the application layer." (Claim 14). This layer consists of code that dynamically generates the presentation portion when a function is requested. At this point decisions may be made regarding destination browser, HTML vs. WML, specific user, etc. This layer is application layer independent.

25. In contrast Lee teaches, "In Step 245, the generator program 28 generates the presentation tier of the application. In this step, the presentation controller classes 162 are created for each entity 50 needed for the generation application including, in a preferred embodiment, security and reports. In Step 250, the generator program 28 creates the business object code 164 for each entity 50 used in the generated software application 40." ([0065]:12-19) With presentation controller classes are generated for each entity, the presentation in whole or in part is included per entity.

E. Claim 1(c)(iii)

"iii. a core layer"

26. The present invention requires "core layer defines the operation of the system itself, whereby changing the core layer results in providing system wide functionalities that affect all applications." (Claim 18). This layer is system wide and not entity specific.

27. In contrast Lee teaches, "the generator program 28 generates a persistence tier 158. In this step, persistence controller classes 160 are created for each entity" ([0065]:3-5) Lee's implementation is entity specific.

F. Claim 1(d)

"d. code definition files providing input to the code generator,"

28. The present invention teaches, "First, the developer creates a code definition file 160 that directs the code generator 170 to generate, in the Oracle embodiment, an Oracle PL/SQL source code file." ([0140]) These files detail the application layer level parameters per entity. The code definition file format is novel.

29. In contrast Lee teaches "a design application configured to receive a system design and create a design database file, and a generator application configured to receive the design database file and create a computer-generated software application that includes a presentation tier, a business tier and a data tier." ([0009]:4-9). Also, "The system of claim 14, wherein said generator application is configured to convert said design database file into an extensible markup language file." (Lee Claim 17)

G. Claim 1(e)

"e. data definition files for defining said data tables, wherein said code generator generates code for said application by processing code definition files, wherein said data definition files configure said data tables to support said toolkit and said application, wherein said data tables comprise user data and operational data for said system, whereby the operation of a plurality of portions of said application can modified by making a single modification to said code definition files."

30. The present invention uses for example a create table statement file as "data definition file". The tables are generated in an Oracle database from a set of files containing Oracle create table, sequence, indexes, triggers, etc.

31. In contrast, Lee uses a different paradigm, i.e. different means.

H. Claim 2:

"The system of claim 1, further comprising at least one document generator, wherein said document generator generates documentation of the design details of the system in at least one document format." (emphasis added)

32. Claim 2 requires a document generator that generates design documentation that details the design and implementation and, for example, includes, entities, attributes, relationships and functions. These are fully indexed with a table of content. In addition, HTML, RTF and other document formats may be generated also as a post process.

33. See the following for details: Abstract lines 21-25, Figure 1, Figure 2, Figure 3, [0012]:6-8, [0012]:23-27, paragraphs [0148], [0149], and [0150].

34. In contrast Lee teaches, "These steps allow the designer to define the type of presentation used by the generated software application 40. The presentation code generated produces the GUI that allows the end user to interact and use the generated software application 40. In a preferred embodiment, the presentation options, including the HTML and GUI layouts, selected by the designer determine the presentation code that is generated in the generated software application 40. Similarly, the entity 50, attributes 64, relationships 66 and lookups 76 defined by the designer using the design program 26 determine the business rules of the system design and affects the business and data code that will be generated. Thus, through the herein described steps, a designer using the software development tool 10 of the present invention can generate a multi-tiered software solution to a system design that integrates and allows interaction between presentation code, business code and data code." (Lee [0058])

35. However Lee fails to teach the required element. There is no discussion of generating documentation outlining the design details of the system. There is discussion of

HTML and GUI layouts of the generated system (application software pages), but no discussion of system documentation generation. These are not a design document as required by the claim.

I. Claim 3:

"The system of claim 1, wherein said network is the Internet."

36. Claim 3 is dependent on claim 1 and thus should be patentable for all the same reasons that claim 1 is patentable over Lee.

J. Claim 4:

"The system of claim 3, further comprising:
a. a web site, connected to the Internet, comprising:
i. a web server in communication with said toolkit,
ii. a file system in communication with said toolkit,
b. at least one remote web browser running on a web browsing device connected to the Internet, wherein said system generates dynamic web pages base on data and programs stored in said database, whereby a user can interact with said application and view said web pages.:

37. Claim 4 requires a web based application that requires a web server, file system, database and browsers in communication the toolkit of claim 1. See Figure 1, Figure 2, Figure 3 and paragraphs [0037], [0040], [0045]:1-4, [0049]:1-3, [0051], [0052], and [0055]. Regarding the file system, the present invention teaches virtual directories, file directories [0064 and 0070]. The present invention teaches web browser template files and dynamic reports in HTML [0066, 0057, 0094]

38. In contrast Lee teaches a web server accesses generated system files ([0072]:17-19), generation of web browser template files ([0066]:1-2), that the user can customize the presentation layer (layout of the web elements) ([0057]:1-14), and reporting support provided by the system ([0095]:1-2)

39. Lee does not teach all the elements required by the claims.

K. Claim 5

"The system of claim 4, further comprising a second web browser being viewed by a second user, wherein said web pages can be customized for specific users, such that the dynamic web page generated for the first user is different than the dynamic web page generated for the second user."

40. The present invention teaches that the web sites are generated completely dynamically allowing for a check to be performed on which user is accessing the application and in return display user specific HTML or content – even if the exact same page is requested.

41. In contrast Lee teaches, "Another aspect of the present invention is the use of dynamic web pages and hook technology. Normally, dynamic web pages are implemented using server-side coding languages such as active server page (ASP) or PHP hypertext preprocessor (PHP). This code is written on text files and placed on a web server. When a reference is made to a server-side code page by a normal web page, the server is expected to be capable of processing this ASP or PHP code." ([0086]) Lee generates static ASP compatible HTML pages. Static pages have embedded queries. Lee's disclosure would only allow for different pages to be displayed if they were already generated for a given user and accessed separately.

L. Claim 6:

The system of claim 4, wherein the system generates web pages for a plurality of formats.

42. In the present invention, "The application programs reference the interface programs to generate dynamic web pages that have a common look and feel," ([0012]) Web pages are totally dynamic.

43. In contrast Lee uses preexisting ASP pages (unlike our pages which are fully dynamically generated). "Another aspect of the present invention is the use of dynamic web pages and hook technology. Normally, dynamic web pages are implemented using server-side

coding languages such as active server page (ASP) or PHP hypertext preprocessor (PHP). This code is written on text files and placed on a web server. When a reference is made to a server-side code page by a normal web page, the server is expected to be capable of processing this ASP or PHP code." ([0086]) Lee generates static ASP compatible HTML pages. Lee's static pages have embedded queries.

M. Claim 7:

"The system of claim 6, wherein said format is for a conventional web browser."

44. In the present invention the web sites are generated completely dynamically, when requested by browser, allowing for a check to be performed on which browser is being used and in return display browser specific or friendly HTML – even if the exact same page is requested. Also, "The system of claim 6 wherein said format is for a mobile device, such as a mobile phone or personal digital assistant." (Claim 8)

45. In contrast Lee teaches, "Another aspect of the present invention is the use of dynamic web pages and hook technology. Normally, dynamic web pages are implemented using server-side coding languages such as active server page (ASP) or PHP hypertext preprocessor (PHP). This code is written on text files and placed on a web server. When a reference is made to a server-side code page by a normal web page, the server is expected to be capable of processing this ASP or PHP code." ([0086]) Lee generates static ASP compatible HTML pages. Lee's static pages have embedded queries.

N. Claim 9

"The system of claim 4 further comprising a predetermined set of code definition files and data definition files, wherein said set of definition files provides a fully functional web site, comprising:
a. default data tables,
b. user interface pages,

c. graphics,
d. toolkit programs providing commonly needed features, such as user accounts, password management, web site administration, billing, and security, whereby a substantially robust web site application is provided without modification of said definition files."

46. The present invention teaches a preprogrammed, initial working web site:

"the toolkit comes complete with an initial working database model (that manages those functions common to most web sites) and the web site user interface pages including graphics, code and pre-programmed modules that can be used as is or customized. They are included in the project and are fully functional. ([0036]:13-18). This includes the following:

> [0085] The system includes a number of existing object modules that perform a variety of complex functions typically found in any web application. These include:
> [0086] User registration/membership,
> [0087] User roles,
> [0088] Secure access to the application, login and "forgot my password" services,
> [0089] Sessions without the use of cookies,
> [0090] Functions and privileges,
> [0091] Dynamic menus (based on user role, logged in, logged out), including bulletin and message notification,
> [0092] Administrative tools to allow for web site management
> [0093] Site-wide bulletins and broadcasts,
> [0094] Inter user messaging,
> [0095] E-mail,
> [0096] Writing files to disk,
> [0097] E-commerce functions that include a store, shopping cart, invoicing, checkout, credit card payment processing and receipts.

47. In contrast Lee does not teach a preexisting application. Instead, Lee provides application design GUI tool for defining entities, attributes and relationships: Fig. 4a, Fig. 4b, Fig. 5, Fig. 6a, Fig. 6b, Fig. 6c, Fig. 7a, Fig. 7b, Fig. 7c, Fig. 7d, Fig. 7e, Fig. 7f, Fig. 7g, Fig. 7.

O. Claim 10

"The system of claim 9 wherein said predetermined set of files provides a working example of how to generate an application such as said robust website application, wherein said working

example provides a starting point for developing a substantially different application by modification of said definition files."

48. The present invention teaches a preprogrammed, initial working web site: "the toolkit comes complete with an initial working database model (that manages those functions common to most web sites) and the web site user interface pages including graphics, code and pre-programmed modules that can be used as is or customized. They are included in the project and are fully functional. ([0036]:13-18). See also: [0085]-[0097].

49. In contrast Lee teaches graphical user interface for HTML interface parameters, "In Step 150, the designer is prompted to select a theme from the hypertext markup language (HTML) options screen." ([0056]:4-6)

P. Claim 11

"The system of claim 1 wherein said code generator, said data database, and the interface and operation of said application can be customized by modifying said code definition files and data definition files."

50. Claim 11 is dependent on claim 1 and thus should be patentable for all the same reasons that claim 1 is patentable over Lee.

Q. Claim 12:

"The system of claim 1 wherein said database contains the structure for said data tables and the data stored in said tables."

51. The present invention teaches a preprogrammed, initial working web site that includes tables with data that support and include the following:

> [0085] The system includes a number of existing object modules that perform a variety of complex functions typically found in any web application. These include:
> [0086] User registration/membership,
> [0087] User roles,
> [0088] Secure access to the application, login and "forgot my password" services,
> [0089] Sessions without the use of cookies,
> [0090] Functions and privileges,

[0091] Dynamic menus (based on user role, logged in, logged out), including bulletin and message notification,
[0092] Administrative tools to allow for web site management
[0093] Site-wide bulletins and broadcasts,
[0094] Inter user messaging,
[0095] E-mail,
[0096] Writing files to disk,
[0097] E-commerce functions that include a store, shopping cart, invoicing, checkout, credit card payment processing and receipts.

52. In contrast Lee does not teach the require structure to support these types of services.

R. Claim 13

"The system of claim 1 wherein one of said application layer, said interface layer, and said core layer can be changed without changing the remaining two layers, whereby such change results in a difference in said application."

53. Claim 13 requires that the Toolkit architecture has 3 distinct layers. Changes may be made to some layers that affect other layers without the need to regenerate.

54. In contrast Lee fails to teach separate source code architecture layers. Lee at 0068 merely refers to customizing the application which is not the same as claim 13.

S. Claim 14:

"The system of claim 13 wherein said interface layer defines a plurality of looks and feels whereby one look and feel can be changed though out the entire application without changing the remaining looks and feels and without changing the application layer."

55. The present invention teaches, "The application programs 130 (collectively known as the web application) relate to each function that will be run on the web site. These functions defer to the interface layer for presentation on the web site." ([0048]) This layer consists of code that dynamically generates the presentation portion when a function is requested. At this point decisions may be made regarding destination browser, HTML vs. WML, specific user, etc. This layer is application layer independent.

56. In contrast Lee teaches "a web browser that accesses the application may represent the presentation tier of the system," ([0004]:10-12) "FIG. 8 is a screen shot that illustrates the GUI a designer uses to define the settings for the appearance of a presentation tier in accordance with an embodiment of the present invention." ([0018]) Lee does not explicitly state that the presentation tier is physically separate from other tiers. He only shows the configuration as a separate step, but not the implementation. Fig. 8 is a GUI that accepts parameters for presentation. These can be implemented in many ways, one of which is through the use of static HTML pages with the options embedded. Lee does not detail exactly how these presentation options are implemented.

T. Claim 17:

"The system of claim 13 wherein said application layer defines the operation of the application, whereby changing the application layer results in a different functional application."

57. Claim 17 is dependent on claim 13 and thus should be patentable for all the same reasons that claim 13 is patentable over Lee.

U. Claim 18:

"The system of claim 13 wherein said database contains a plurality of customizations that result in a plurality of substantially different applications, and wherein the core layer defines the operation of the system itself, whereby changing the core layer results in providing system wide functionalities that affect all applications."

58. The present invention requires "The system of claim 1 wherein one of said application layer, said interface layer, and said core layer can be changed without changing the remaining two layers, whereby such change results in a difference in said application" (Claim 13) for example showing independence of layers.

59. In contrast, Lee only teaches "computer-generated software application includes a presentation tier, a business tier and a data tier." (Claim 14:11-13). There is no

mention of a core layer equivalent "tier". Lee at 0068 merely talks about customizing, but not specifically how.

60. Lee's tiers are the same as applicants' layers. Lee does not teach or suggest the required elements of claim 18.

V. Claim 20(a)

"In a system comprising of a web server, a database, a toolkit, and an application code generator, a method of producing custom web sites with substantially different looks and operations for diverse business disciplines comprising the steps of:

a. defining data in data definition files, ..."

61. In the present invention data definition files are, for example, Oracle SQL files that contain create table statements.

62. In contrast Lee teaches, "In a preferred embodiment, the design database file 34 created by the design program 26 is passed to the generator program 28 where it is reformatted as an extensible markup language (XML) meta document." ([0036]:1-4) This design database file is converted to XML and created by the design program.

W. Claim 20(b)

"b. specifying code in code definition files,

63. In the present invention the code generator uses code definition files to generate database programs, for example PL/SQL code as stored procedures in an Oracle database.

64. In contrast, Lee uses a different paradigm, i.e. different means.

X. Claim 20(c)

"c. generating data tables based on said data definition files,

65. In the present invention, tables are generated, for example, in an Oracle database from a set of files containing Oracle create table, sequence, indexes, triggers, etc.

66. In contrast, Lee uses a different paradigm, i.e. different means.

Y. Claim 20(d)

"d. generating code based on said code definition files,"

67. In the present invention, the code generator uses code definition files to generate database programs , for example PL/SQL code as stored procedures in an Oracle database.

68. In contrast, Lee uses a different paradigm, i.e. different means.

Z. Claim 20(e)

"e. storing said generated code in said database along with said toolkit, and..."

69. In the present invention the system may, for example work, completely from PL/SQL code as stored procedures in an Oracle database.

70. In contrast, Lee uses a different paradigm, i.e. different means.

71. Claim 20(f)

f. modifying data stored in said database associated with at least one of said web sites, whereby said system will dynamically generate web pages for at least one of said websites having a substantially different look or operation than at least one other of said websites.

72. In the present invention, the system is completely dynamic, allowing even for data (parameter values) to govern how pages are generated, look and feel.

73. In contrast, Lee uses a different paradigm, i.e. different means.

AA. Claim 5:

"The system of claim 4, further comprising a second web browser being viewed by a second user, wherein said web pages can be customized for specific users, such that the dynamic web

page generated for the first user is different than the dynamic web page generated fro the second user."

74. Lee does not mentiong this. Twaddle teaches and claims, "A method of dynamic web-page generation comprising steps of: retrieving a user template from a database, said template containing one or more web page component data structures; parsing the retrieved user template sequentially to determine the taxonomy of the hierarchical references; and compiling at least a pointer to each referenced page component data structure sequentially in accordance with the taxonomy of the hierarchical references to create a web-page containing at least one pointer to each of the referenced data structures, wherein each page component data structure is one of a locally stored pre-created structure, a dynamically created structure, or remotely stored structure." (Claim 1)

75. However the combination of Lee and Twaddle does not teach or suggest every element of applicants' invention.

BB. Claim 8:

"The system of claim 6 wherein said format is for a mobile device, such as a mobile phone or personal digital assistant."

76. In the present invention, the interface layer consists of code that dynamically generates the presentation portion when a function is requested. At this point decisions may be made regarding destination browser, HTML vs. WML, specific user, etc. These decisions are made as a function is requested. This layer is application layer independent.

77. Lee does not mention phone, WAP or WML. Lee generates static HTML pages with ASP hooks. Further Mansour does not mention WML nor any mention of dynamically generating pages for specific users.

78. Thus neither Lee nor Mansour nor their combination teach or suggest the required elements of applicants' invention.

CC. Claim 19:

"The system of claim 18, further comprising at least one Internet server configured for hosting a substantially large number of applications, wherein said server is operated by an Internet Service Provider providing services to a plurality of application owners, whereby each of said application owners share a common application or core layer and is provided a custom look and feel for their specific application by customization of the data in the database that affects the interface."

79. The present invention teaches, for example, a single server hosting a plurality of applications. This is facilitated, for example, through Oracle's accounts and grants for use of other accounts tables, data, triggers, procedures, etc. This is also facilitated through the multi-layer architecture keeping the core layer separate from other layers.

80. Neither Lee nor Mansour teach or suggest this type of architecture.

81. Thus neither Lee nor Mansour nor their combination teach or suggest the required elements of applicants' invention.

DD. Claim 15:

"The system of claim 14, wherein different formats are generated by the toolkit for display on different web browsing devices, whereby one said change results in changing the look and feel of what is displayed on a plurality of web browsing devices. "

82. In the present invention, the interface layer consists of code that dynamically generates the presentation portion when a function is requested. At this point decisions may be made regarding destination browser, HTML vs. WML, specific user, etc. These decisions are made as a function is requested. This layer is application layer independent.

83. Neither Lee nor Mansour teach the dynamic pages of the present invention. Twaddle teaches the following:

[0051] Once the portal page builder has compiled the web-page source code the source code is passed to the web server for transmission over the Internet to the user 114, who displays the source code using a standard web browser.

[0052] The compiled source code generated by the portal page builder module 104 may be in any mark-up language, but in particular HTML, XML, and WML are envisaged.

[0053] In addition, as an alternative to including a pointer to an external content supplier 112 in the form of a URL, it may be possible for the portal page builder to generate a page request which is transmitted by the web server 102 over the Internet 100 to the content supplier's website, which then provides the requested content via the Internet 100 and web server 102, and the portal page builder module 104 incorporates the content direct into the compiled source code. Further details of this alternative operation are described later.

84. As proposed by the examiner, the combination of Lee and Mansour and Twaddle would not necessarily produce totally dynamic system.

EE. **Claim 16:**

"The system of claim 15, wherein one of said plurality of web browsing devices support the WAP format."

85. In the present invention the interface layer consists of code that dynamically generates the presentation portion when a function is requested. At this point decisions may be made regarding destination browser, HTML vs. WML, specific user, etc. These decisions are made as a function is requested. This layer is application layer independent.

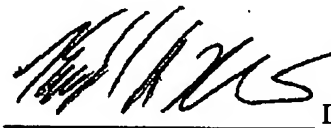
86. As proposed by the examiner, the combination of Lee and Mansour and Twaddle would not necessarily produce totally dynamic system.

IX. CONCLUSION

The undersigned respectfully submits that, in view of the foregoing remarks, the present application is believed to be in condition for allowance. It is respectfully requested that this application be considered and that this case be passed to issue. If it is believed that a telephone

conversation would expedite the prosecution of the present application, or clarify matters with regard to its allowance, the Examiner is invited to call the undersigned at 408-739-9517.

Respectfully submitted,



Date: December 28, 2006

Kendyl A. Román
730 Bantry Court
Sunnyvale, CA 94087
Phone: 408-739-9517